Assignment #1
CSCI 201 Spring 2017
2.5% of course grade

Over the course of the semester you will be developing a social media application to share opinions about and rate movies between users. For this assignment, all functionality will be done through the console. You will be utilizing your knowledge of Java, and file parsing to complete this. You will also be independently learning how to parse an xml file.

The xml file will contain all the information that the application needs. It will contain the following sections described, **but not in any particular order**. The relative file path will be passed in as a command line argument. At the end of these instructions is a quick tutorial on how to add a command line argument in Eclipse.

The file will contain a list of possible genres that each movie can have.

```
<genres>
   <genre>Animated</genre>
   <genre>Drama</genre>
   <genre>Action</genre>
   <genre>Thriller</genre>
   <genre>Comedy</genre>
   <genre>Romance</genre>
</genres>
```

It will also list the actions that are possible as a user to perform (in association with a movie).

```
<actions>
   <action>Watched</action>
   <action>Liked</action>
   <action>Disliked</action>
   <action>Rated</action>
</actions>
```

It will list the movies stored in the application. Each movie will have the title, director, writer(s), actor(s), genre, description and the year it was released. It could also have an average rating, but this field is optional.

```
<movies>
   <movie>
         <title>How to Train Your Dragon</title>
         <director>Dean DeBlois</director>
         <writers>
               <writer>Dean DeBlois</writer>
               <writer>William Davies</writer>
               <writer>Chris Sanders</writer>
         </writers>
         <year>2010</year>
         <genre>Animated</genre>
```

```
        <description>A hapless young Viking who aspires to hunt
   dragons becomes the unlikely friend of a young dragon himself, and
   learns there may be more to the creatures than he assumed.</
   description>
        <rating></rating>
        <actors>
             <actor>Jay Baruchel</actor>
             <actor>Gerard Butler</actor>
             <actor>America Ferrera</actor>
        </actors>
   </movie>
</movies>
```

And of course, it will list the users. A user's profile will contain their first and last name, their username and password, the users they are 'following', and a 'feed' which contains all of their events.

```
<users>
     <user>
          <username>knguyen</username>
          <password>knguyen</password>
          <fname>Kien</fname>
          <lname>Nguyen</lname>
          <following>
               <username>jmiller</username>
          </following>
          <feed>
               <event>
                    <action>Liked</action>
                    <movie>Dirty Rotten Scoundrels</movie>
                    <rating></rating>
               </event>
          </feed>
     </user>
     <user>
          <username>jmiller</username>
          <password>jmiller</password>
          <fname>Jeffrey</fname>
          <lname>Miller</lname>
          <following></following>
          <feed>
               <event>
                    <action>Disliked</action>
                    <movie>Shutter Island</movie>
                    <rating></rating>
               </event>
          </feed>
     </user>
</users>
```

Before you begin parsing the xml file by hand, it is suggested that you first look into using one of Java's built in xml parsers such as DOM or SAX. Below are some links to information on these options.

https://docs.oracle.com/javase/tutorial/jaxp/dom/
https://www.tutorialspoint.com/java_xml/java_dom_parser.htm
http://tutorials.jenkov.com/java-xml/sax-example.html
https://docs.oracle.com/javase/tutorial/jaxp/sax/parsing.html
https://www.tutorialspoint.com/java_xml/java_sax_parse_document.htm

You will need to store the information from the file in order to allow for the following functionalities. The application will only have two options in the main menu which will be to login or to exit. We will not have the option to create an account as you have not yet learned about data persistence. For now, the only users will be the ones stored in the file.

```
1. Login
2. Exit
```

If the user enters something other than '1' or '2' (i.e. another number or a string value), they should see a message like the following:

```
You have entered an invalid command, please try again.
```

At which point you should print the menu again. Continue this process until they have entered a valid command.

If they choose to exit, the program should terminate. If they choose to login, they should be queried for their username. If the user provides a non-existing username, they should be queried 2 more times, and if each entry is invalid, go back to the main menu. Example ('->' represents the user's input):

```
Please enter your username
-> elautz
Invalid username. You have 2 more chances to enter a valid username.
-> elautz
Invalid username. You have 1 more chance to enter a valid username.
-> elautz
1. Login
2. Exit
```

If they enter a valid username, they should be queried for their password. If the password does not match the username, the user gets 2 more tries to provide the correct one. If they fail to do so, they should be taken back to the main menu. This should look similar to the process of entering an incorrect username.

```
Please enter your username
-> knguyen
```

```
Please enter your password.
-> elautz
Incorrect password. You have 2 more chances to enter the correct
password.
```

If the user provides the appropriate password, they should be shown another menu (let's refer to this as the login menu) with the following options:

```
1. Search Users
2. Search Movies
3. View Feed
4. View Profile
5. Logout
6. Exit
```

If the user enters something other than '1', '2', '3', '4', '5', or '6' (i.e. another number or a string value), they should see a message that they entered an invalid command (just like on the main menu), and the login menu should print again.

If the user chooses to logout (5), they should be taken back to the main menu. If they choose to exit (6), the program should terminate.

Again, since you have not learned about data persistence, the rest of the menu options focus on viewing the information in the application.

If they choose to view their profile (4), there should be a printout with all of the user's information. They should see their full name, username, and password with only the first and last character visible. They should see a list of the users they are following, **and it should show who is following them.** They should also see a message that if they would like to return to the (login) menu, to type '0'. Again, if they input anything other than '0', they should see a message that their input was invalid, and queried again. Below is an example (logged in as knguyen):

```
1. Search Users
2. Search Movies
3. View Feed
4. View Profile
5. Logout
6. Exit
->4
Kien Nguyen
username: knguyen
password: k*****n
Following:
   jmiller
Followers:

To go back to the login menu, please type '0'
->1
```

```
Invalid command. To go back to the login menu, please type '0'
->0

1. Search Users
2. Search Movies
3. View Feed
4. View Profile
5. Logout
6. Exit
```

If they choose to view their feed (3), it should show all the events in their feed, and all the events in the feeds of the users they are following. They should also see a message that if they would like to return to the menu, to type '0'. Below is an example (again logged in as knguyen):

```
1. Search Users
2. Search Movies
3. View Feed
4. View Profile
5. Logout
6. Exit
->3
Feed:

knguyen liked the movie 'Dirty Rotten Scoundrels'.
jmiller disliked the movie 'Shutter Island'.

To go back to the login menu, please type '0'
```

If they choose to search for a user (1), they should be queried for the username they are searching for. The search should NOT be case sensitive. For example, if knguyen searches for 'JMILLER', the search should return 'jmiller' as a result.

They should also see a message that if they would like to return to the menu, to type '0'.Below is an example (again logged in as knguyen):

```
Please enter the username you are searching for.
->JMILLER
1 result:
jmiller
```

At this point they should have two options: search again for another user, or return to the login menu. If they enter any other input than '1' or '2' they should see a message that their input was invalid, and the menu should print again.

```
Please choose from the following options:
1. Back to Login Menu
2. Search Again
```

```
->2
Please enter the username you are searching for.
->elautz
0 results

Please choose from the following options:
1. Back to Login Menu
2. Search Again
->1
```

If the user chooses to search for a movie (2), they should be taken to the movie search submenu:

```
1. Search by Actor
2. Search by Title
3. Search by Genre
4. Back to Login Menu
```

If they enter any other input than '1', '2', '3', or '4' they should see a message that their input was invalid, and the menu should print again. If they choose option '4', the login menu should be displayed again.

If they choose any of the search options they should see a message querying them for either the genre, actor or title they wish to search by. All these searches should return a list of movie titles. Like the user search, the search should not be case sensitive. Once the search results are displayed, they should see a similar menu to the one shown after a user search (in the following example, we will search by actor):

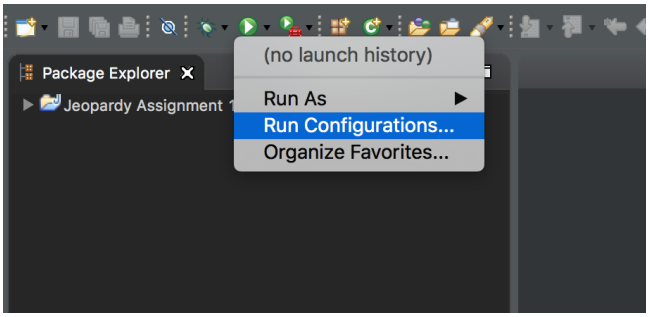
```
1. Search by Actor
2. Search by Title
3. Search by Genre
4. Back to Login Menu
->1
Please enter the name of the actor you wish to search by.
->JAY BARUCHEL
1 result:
'How to Train Your Dragon'

Please choose from the following options:
1. Back to Login Menu
2. Back to Search Movies Menu
```

Again, if any input is given other than '1' or '2', they should see a message that their input was invalid, and the menu should print again. If they choose option '1', the login menu should print, and if they choose option '2' the search menu should print (search by actor, title or genre).

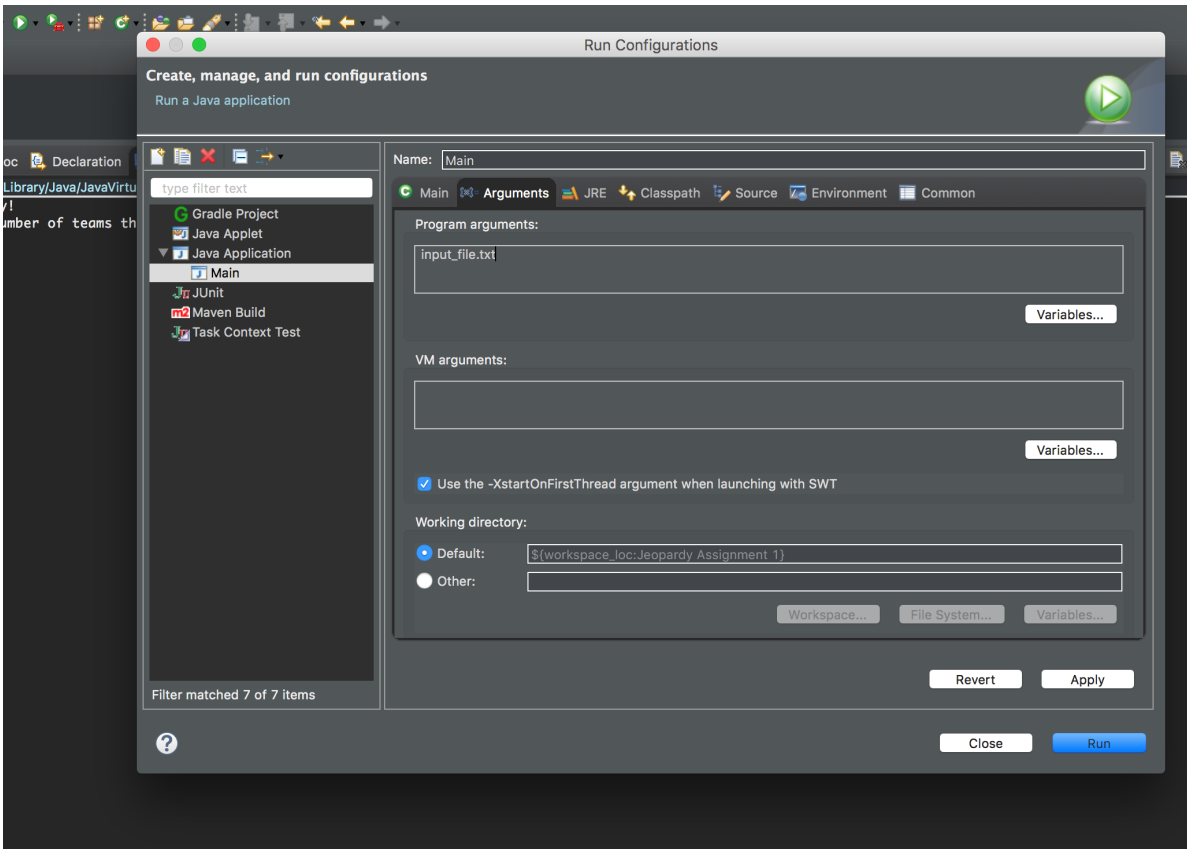
For this assignment, you do not need to handle any file format errors.

Below is the tutorial for inputing command line arguments in Eclipse. After that, you will see the break down of the points for this assignment.

In Eclipse, if you click either the project folder, or the class with your main method, and then click the run button (green circle with white triangle), a run configuration will be generated. Then, if you click the small black down arrow next to the run button you should see the following menu. Click 'Run Configurations'.



Then you will see a similar screen to the one shown below. Under 'Java Application' on the left hand side, you should see a configuration labeled with the same name as your class with the main method (in the example, the class is named 'Main'). If you click on the configuration, you will see a box on the right labeled 'Program Arguments'. This is where you will enter the relative path to the xml file you are testing with. In Eclipse, the top directory is the project



folder. So, for example, in the screenshot above, since the program argument is 'input_file.txt', it is located directly in the project folder.

Grading Criteria Total (2.5%)

XML Parsing: 0.9%

- 0.05% Genres are properly parsed
- 0.05% Actions are properly parsed
- 0.4% User information is properly parsed
- 0.4% Movie information is properly parsed

Searching (0.5%)

- 0.1% Searching for a user returns all expected usernames
- 0.1% Searching for movies by title returns all movies with that title
- 0.2% Searching for movies by genre returns all movies with that genre
- 0.2% Searching for movies by actor returns all movies with that actor

Viewing User's Information (0.6%)

- 0.2% The user's profile printout contains accurate information
- 0.15% The user's profile printout accurately displays their followers
- 0.05% The user's profile printout only shows the first and last characters of their password
- 0.2% The user's feed printout shows all their events and all events from the users they are following

Other (0.5%)

- 0.05% The xml file path is passed is as a command line argument
- 0.05% All searches are not case sensitive.
- 0.2% All invalid inputs are handled on each menu
- 0.2% All menu options navigate to the expected submenu or printout.